

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**A METHOD AND APPARATUS FOR SHOT DETECTION**

Inventors:

**Wei Qi  
Dong Zhang**

ATTORNEY'S DOCKET NO. MS1-717US

1            **TECHNICAL FIELD**

2            This invention generally relates to image processing and analysis and, more  
3            particularly, to a method and apparatus for shot detection.

5            **BACKGROUND**

6            With recent improvements in processing, storage and networking  
7            technologies, many personal computing systems have the capacity to receive,  
8            process and render multimedia objects (e.g., audio, graphical and video content).  
9            One example of such computing power applied to the field of multimedia  
10          rendering, for example, is that it is now possible to “stream” media content from a  
11          remote server over a data network to an appropriately configured computing  
12          system for rendering on the computing system. Many of the rendering systems  
13          provide functionality akin to that of a typical video cassette player/recorder (VCR).  
14          However, with the increased computing power comes an increased expectation by  
15          consumers for even more advanced capabilities. A prime example of just such an  
16          expectation is the ability to rapidly identify, store and access relevant (i.e., of  
17          particular interest to the user) media content. Conventional media processing  
18          systems fail to meet this expectation.

19          In order to store and/or access a vast amount of media efficiently, the media  
20          must be parsed into uniquely identifiable segments of content. Many systems  
21          attempt to parse video content, for example, into *shots*. A shot is defined as an  
22          uninterrupted temporal segment in a video sequence, and often defines the low-  
23          level syntactical building blocks of video content. Shots, in turn, are comprised of  
24          a number of frames (e.g., 24 frames-per-second, per shot). In parsing the video  
25          into shots, conventional media processing systems attempt to identify shot

1 boundaries by analyzing consecutive frames for deviations in content from one  
2 another. A common approach to distinguish content involves the use of color  
3 histogram based segmentation. That is, generating a color histogram for each of a  
4 number of consecutive frames and analyzing the histogram difference of  
5 consecutive frames to detect a significant deviation. A deviation within a single  
6 frame that exceeds a deviation threshold is determined to signal a shot boundary.  
7

8 While the use of color histogram may prove acceptable in certain limited  
9 circumstances, it is prone to false shot boundary detection in many applications.  
10 Take, for example, news footage. News footage often includes light flashes from  
11 camera flash bulbs, emergency vehicle lights, lightning from a storm, bright stage  
12 lights for the video camera equipment and the like. The result is that one shot of  
13 such news footage may include a number of light flashes (flashlight phenomena)  
14 which conventional shot boundary detection schemes mistake for shot boundaries.  
15 Another example of media including the flashlight phenomena is action and  
16 science fiction movies, sporting events, and a host of other media – media for  
17 which conventional shot detection schemes are ill-suited.

18 The challenge of distinguishing flashlight phenomena from actual shot  
19 boundaries is not trivial. The limitation of conventional shot boundary detection  
20 schemes is that they assume a flashlight only occurs across a single frame. In the  
21 real world, not only can flashlights span multiple frames, they can span a shot  
22 boundary.

23 Another limitation of such conventional shot boundary detection schemes is  
24 that of threshold selection, i.e., the threshold of, for example, color histogram  
25 deviation that signals a shot boundary. Many conventional shot boundary  
detection schemes use global, pre-defined thresholds, or simple local window

1 based adaptive thresholds. Global thresholds generally provide the worst  
2 performance as video properties often vary a lot and, quite simply, one size  
3 (threshold) does not fit all. The local window based adaptive threshold selection  
4 method also has its limitations insofar as in certain situations, the local statistics  
5 are polluted with strong noises such as, for example, loud noises and/or flashlight  
6 effects.

7 Thus, a method and apparatus for shot boundary detection is presented,  
8 unencumbered by the inherent limitations commonly associated with prior art  
9 systems.

10

## 11 **SUMMARY**

12 A method and apparatus for shot boundary detection are described herein.  
13 In accordance with a first example embodiment, a method identifies an abrupt  
14 transition in content between two frames, and determines whether the abrupt  
15 transition was caused by a shot boundary between the two frames or by a flashlight  
16 event is presented.

17 In one implementation, identifying an abrupt transition includes calculating  
18 a difference in light intensity histograms between the current frame and a  
19 preceding frame, and comparing the histogram difference to a dynamically  
20 determined threshold, such that an abrupt transition is indicated if the histogram  
21 difference exceeds the threshold.

22 In another embodiment, a shot boundary detector statistically analyzes one  
23 or more attributes associated with content in video frames to detect abrupt and/or  
24 gradual transitions in the video content indicative of a shot boundary. A flashlight  
25 detector, responsive to the shot boundary detector, distinguishes abrupt transitions

in the video content caused by flashlight events from those caused by actual shot boundaries, to reduce false-positive identification of flashlight events as shot boundaries.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

The same reference numbers are used throughout the figures to reference like components and features.

**Fig. 1** is a block diagram of an example computing system incorporating the teachings of the present invention;

**Fig. 2** is a block diagram of an example media analysis agent to identify shot boundaries, in accordance with one example embodiment of the present invention;

**Fig. 3** is a graphical illustration distinguishing the shot cut model from the flashlight model, according to one aspect of the present invention;

**Fig. 4** is a graphical illustration of an example data structure to store frames of an identified shot, according to one embodiment of the present invention;

**Fig. 5** is a flow chart of an example method of shot boundary detection, according to one embodiment of the present invention;

**Fig. 6** is a flow chart of an example method of distinguishing a cut model from a flashlight model in shot boundary analysis, according to one aspect of the present invention;

**Fig. 7** is a flow chart of an example method of dynamically generating an adaptive threshold to identify shot boundaries in a video sequence, according to one aspect of the present invention; and

1       **Fig. 8** is a block diagram of an example storage medium having stored  
2 thereon a plurality of executable instructions including at least a subset of which  
3 that, when executed, implement a media analysis agent incorporating the teachings  
4 of the present invention.

5

6 **DETAILED DESCRIPTION**

7       This invention concerns a method and apparatus for shot detection. In this  
8 regard, the present invention overcomes a number of the limitations commonly  
9 associated with the prior art image storage and retrieval systems. The inventive  
10 nature of the present invention will be developed within the context of visual  
11 media content such as, for example, video media content. It is to be appreciated,  
12 however, that the scope of the present invention is not so limited. The innovative  
13 media analysis agent introduced below may well utilize the inventive concepts  
14 described herein to perform media segmentation on any of a wide variety of  
15 multimedia content including, for example, audio content, graphical content, and  
16 the like. In this regard, the example embodiments presented below are merely  
17 illustrative of the scope and spirit of the present invention.

18       In describing the present invention, example network architectures and  
19 associated methods will be described with reference to the above drawings. It is  
20 noted, however, that modification to the methods and apparatus described herein  
21 may well be made without deviating from the present invention. Indeed, such  
22 alternate embodiments are anticipated within the scope and spirit of the present  
23 invention.

1            **EXAMPLE COMPUTING SYSTEM**

2            **Fig. 1** illustrates an example computing system 102 including an innovative  
3 media analysis agent 104, to analyze media content, identify and segment the  
4 content into shots based, at least in part, on a statistical analysis of the light content  
5 of at least a subset of the frames comprising the video content. Unlike the  
6 conventional segmentation systems introduced above, however, the media analysis  
7 agent 104 is able to accurately distinguish flashlight phenomena occurring within a  
8 shot from actual shot boundaries, even if the shot boundary is embedded within the  
9 flashlight effect. In accordance with another aspect of the invention, to be  
10 described more fully below, media analysis agent 104 overcomes the limitations  
11 commonly associated with threshold selection introduced above, by adopting an  
12 adaptive threshold selection technique, to dynamically select a threshold suitable  
13 for the content of the media. It will be evident, from the discussion to follow, that  
14 computer 102 is intended to represent any of a class of general or special purpose  
15 computing platforms which, when endowed with the innovative analysis agent  
16 104, implement the teachings of the present invention in accordance with the first  
17 example implementation introduced above. It is to be appreciated that although  
18 analysis agent 104 is depicted in the context of Fig. 1 as a software application,  
19 computer system 102 may alternatively support a hardware implementation of  
20 agent 104 as well. In this regard, but for the description of analysis agent 104, the  
21 following description of computer system 102 is intended to be merely illustrative,  
22 as computer systems of greater or lesser capability may well be substituted without  
23 deviating from the spirit and scope of the present invention.

1 As shown, computer 102 includes one or more processors or processing  
2 units 132, a system memory 134, and a bus 136 that couples various system  
3 components including the system memory 134 to processors 132.

4 The bus 136 represents one or more of any of several types of bus  
5 structures, including a memory bus or memory controller, a peripheral bus, an  
6 accelerated graphics port, and a processor or local bus using any of a variety of bus  
7 architectures. The system memory includes read only memory (ROM) 138 and  
8 random access memory (RAM) 140. A basic input/output system (BIOS) 142,  
9 containing the basic routines that help to transfer information between elements  
10 within computer 102, such as during start-up, is stored in ROM 138. Computer  
11 102 further includes a hard disk drive 144 for reading from and writing to a hard  
12 disk, not shown, a magnetic disk drive 146 for reading from and writing to a  
13 removable magnetic disk 148, and an optical disk drive 150 for reading from or  
14 writing to a removable optical disk 152 such as a CD ROM, DVD ROM or other  
15 such optical media. The hard disk drive 144, magnetic disk drive 146, and optical  
16 disk drive 150 are connected to the bus 136 by a SCSI interface 154 or some other  
17 suitable bus interface. The drives and their associated computer-readable media  
18 provide nonvolatile storage of computer readable instructions, data structures,  
19 program modules and other data for computer 102.

20 Although the example operating environment described herein employs a  
21 hard disk 144, a removable magnetic disk 148 and a removable optical disk 152, it  
22 should be appreciated by those skilled in the art that other types of computer  
23 readable media which can store data that is accessible by a computer, such as  
24 magnetic cassettes, flash memory cards, digital video disks, random access

memories (RAMs) read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk 144, magnetic disk 148, optical disk 152, ROM 138, or RAM 140, including an operating system 158, one or more application programs 160 including, for example, analysis agent 104 incorporating the teachings of the present invention, other program modules 162, and program data 164 (e.g., resultant language model data structures, etc.). A user may enter commands and information into computer 102 through input devices such as keyboard 166 and pointing device 168. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to the processing unit 132 through an interface 170 that is coupled to bus 136. A monitor 172 or other type of display device is also connected to the bus 136 via an interface, such as a video adapter 174. In addition to the monitor 172, personal computers often include other peripheral output devices (not shown) such as speakers and printers.

As shown, computer 102 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 176. The remote computer 176 may be another personal computer, a personal digital assistant, a server, a router or other network device, a network “thin-client” PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 102, although only a memory storage device 178 has been illustrated in Fig. 1. In this regard, innovative analysis agent 104 may well be invoked and utilized by remote computing systems such as, for example, computing system 176.

As shown, the logical connections depicted in Fig. 1 include a local area network (LAN) 180 and a wide area network (WAN) 182. Such networking environments are commonplace in offices, enterprise-wide computer networks, Intranets, and the Internet. In one embodiment, remote computer 176 executes an Internet Web browser program such as the "Internet Explorer" Web browser manufactured and distributed by Microsoft Corporation of Redmond, Washington to access and utilize online services.

When used in a LAN networking environment, computer 102 is connected to the local network 180 through a network interface or adapter 184. When used in a WAN networking environment, computer 102 typically includes a modem 186 or other means for establishing communications over the wide area network 182, such as the Internet. The modem 186, which may be internal or external, is connected to the bus 136 via input/output (I/O) interface 156. In addition to network connectivity, I/O interface 156 also supports one or more printers 188. In a networked environment, program modules depicted relative to the personal computer 102, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Generally, the data processors of computer 102 are programmed by means of instructions stored at different times in the various computer-readable storage media of the computer. Programs and operating systems are typically distributed, for example, on floppy disks or CD-ROMs. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention

described herein includes these and other various types of computer-readable storage media when such media contain instructions or programs for implementing the innovative steps described below in conjunction with a microprocessor or other data processor. The invention also includes the computer itself when programmed according to the methods and techniques described below. Furthermore, certain sub-components of the computer may be programmed to perform the functions and steps described below. The invention includes such sub-components when they are programmed as described. In addition, the invention described herein includes data structures, described below, as embodied on various types of memory media.

For purposes of illustration, programs and other executable program components such as the operating system are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computer, and are executed by the data processor(s) of the computer.

It should be appreciated that although depicted as a separate, stand alone application in Fig. 1, media analysis agent 104 may well be implemented as a function of a higher level application, e.g., a media player, a media librarian/database, a ripper application, etc.

## EXAMPLE MEDIA ANALYSIS AGENT

**Fig. 2** illustrates a block diagram of an example media analysis agent 104 incorporating the teachings of the present invention, according to one embodiment of the invention. In accordance with the illustrated example embodiment of Fig. 2, media analysis agent 104 is depicted comprising one or more controller(s) 202, a media analysis engine 204, memory/storage device 206, input/output interface(s)

1 208 and, optionally, one or more applications 210, each communicatively coupled  
2 as shown. As introduced above, media analysis agent 104 analyzes one or more  
3 attributes of received frames and, using a dynamically selected threshold, segments  
4 the video content into shots.

5 In accordance with one example implementation of the present invention,  
6 media analysis agent 104 selectively invokes shot boundary detector 212 to  
7 segment the received video content into uniquely identifiable shots. According to  
8 one example implementation, to be developed more fully below, shot boundary  
9 detector 212 analyzes the light histogram difference of successive frames to  
10 identify disparate shots within the sequence of frames. If the histogram difference  
11 reaches a particular threshold, shot boundary detector 212 invokes an instance of  
12 flashlight detector 214, to distinguish a potential flashlight condition from an  
13 actual shot cut boundary. In addition, rather than rely on global pre-defined  
14 threshold(s) (which it could use in alternate embodiments), media analysis engine  
15 204 is depicted comprising an adaptive threshold selection module 216 to  
16 dynamically select thresholds that are representative of the content being analyzed.  
17 Based, at least in part on the analysis of light intensity of the video content, media  
18 analysis engine 204 segments the received media into shots for subsequent  
19 content-based access and retrieval. It is to be appreciated that, although depicted  
20 as a number of disparate functional blocks, one or more of elements 202-220 may  
21 well be combined into one or more blocks, without deviating from the spirit and  
22 scope of the present invention.

23 According to one implementation, controller(s) 202 receive media content  
24 from any of a number of sources including, for example, local memory storage  
25 (206), remote media provider(s) and/or content storage source(s), and audio/video

1 devices communicatively coupled to media analysis agent 104. According to one  
2 implementation, the media content is received from remote sources by controller(s)  
3 202 and placed in storage/memory 218 for analysis. According to one  
4 implementation, the media is received by the host computer 102 in compressed  
5 form and is decompressed before presentation to media analysis agent 104. In an  
6 alternate implementation, controller(s) 202 selectively invoke a decoder  
7 application (e.g., 210) resident within or outside of media analysis agent 104 to  
8 decode media received in compressed form before selectively invoking the media  
9 analysis engine 204.

10 But for the innovative aspects of the invention, described above,  
11 controller(s) 202 is intended to represent any of a wide variety of control logic  
12 known in the art such as, for example, a processor, an application specific  
13 integrated circuit (ASIC), a field programmable gate array (FPGA), a  
14 programmable logic array (PLA), and the like. Moreover, it should be appreciated  
15 that controller(s) 202 may well be implemented in software as a plurality of  
16 executable instructions which, when executed (e.g., by processing unit 132),  
17 implement the control functions described above.

18 Media analysis engine 204 is selectively invoked by controller(s) 202 to  
19 segment media content into uniquely identifiable shots of video content. As  
20 introduced above, media analysis engine 204 includes a shot boundary detector  
21 212 that analyzes the light histogram difference of a sequence of frames to identify  
22 shot boundaries. More particularly, shot boundary detector 212 generates a light  
23 histogram for each frame of video content and analyzes successive frames'  
24 histograms difference for indications denoting a shot change. According to one  
25 example implementation, shot boundary detector 212 implements a twin-threshold

1 detection scheme, wherein one or more statistics generated from the histograms are  
2 compared to two thresholds to identify shot boundaries.

3 According to one implementation, shot boundary detector 212 utilizes a  
4 high threshold, Tb, and a lower threshold, Ts, in the twin-threshold detection  
5 scheme. A high histogram difference threshold (Tb) is set, wherein content that  
6 exceeds this high threshold might include a flashlight or a shot boundary.  
7 Accordingly, in accordance with one aspect of the present invention, shot boundary  
8 detector 212 selectively invokes an instance of the flashlight detector module 214  
9 to determine whether the high light intensity is due to a flashlight condition or an  
10 actual shot boundary.

11 If the histogram difference does not exceed the high threshold, Tb, but does  
12 exceed a lower threshold (Ts), it is regarded as a start frame of gradual transition,  
13 and shot boundary detector 212 analyzes several frames of such content to  
14 determine whether a gradual shot transition is taking place over several frames.  
15 According to one implementation, shot boundary detector 212 accumulates the  
16 histogram difference over two or more frames to see if, after accumulating two or  
17 more frames worth of deviation, the accumulated difference exceeds the high  
18 threshold, Tb. If so, shot boundary detector identifies a gradual transition shot  
19 boundary. If not, the candidate is abandoned and shot boundary detector 212  
20 continues with analysis of the next frame.

21 As introduced above, media analysis engine 104 does not rely on the  
22 conventional practice of global, pre-set thresholds or thresholds that are arbitrarily  
23 set based on the content of a sliding window of frames. Rather, media analysis  
24 engine 204 includes an adaptive threshold selection module 216 incorporates  
25 certain safeguards that ensure that thresholds are set in accordance with the

1 received video content only if the content is relatively stable. According to one  
2 example implementation, adaptive threshold selection module 216 generates a  
3 sliding window of frames preceding the current frame, and analyzes the frames  
4 within the sliding window to generate a statistical model of the light intensity of  
5 the frames within the window. In one implementation, for example, threshold  
6 selection module 216 generates a sliding window of 15-25 frames, and calculates  
7 an average value and standard deviation of histogram differences within the sliding  
8 window. The average value is used to calculate each of the low threshold,  $T_s$ , and  
9 the high threshold,  $T_b$ . According to one example implementation, the low  
10 threshold  $T_s$  is set to two- to three-times the average intensity, while the high  
11 threshold  $T_b$  is set to four- to five-times the average intensity.

12 Unlike convention threshold selection schemes, however, adaptive  
13 threshold selection module 216 verifies the appropriateness of the resultant  
14 thresholds before using the threshold(s) to determine whether the current frame is  
15 a shot boundary. According to one implementation, threshold selection module  
16 216 analyzes the standard deviation of the intensity values within the window to  
17 determine whether the frames within the sliding window present a relatively stable  
18 light intensity model. If, for example, selection module 216 were to calculate the  
19 standard deviation of light intensity over several frames that included several  
20 flashlights, the average intensity would be so high, compared to the other media,  
21 that setting thresholds based upon such content would provide for artificially high  
22 thresholds, perhaps resulting in missed shot boundaries. However, such a window  
23 of several flashlights will be characterized by a relatively high standard deviation,  
24 which is an indicator to threshold selection module 216 that, perhaps, the data is  
25 not a good source from which to develop thresholds. According to one

1 implementation, if the standard deviation is higher than a threshold, that is to say  
2 the distribution of histogram differences within the sliding window is dispersed, a  
3 threshold calculated based on such data is un-trustworthy. The reason being that  
4 the video content in the window is changing too drastically to be of use in  
5 threshold selection. In such a circumstance, threshold selection module 216  
6 reverts to the last acceptable thresholds,  $T_s$  and  $T_b$ . According to another  
7 embodiment, upon determining that the values within the sliding window do not  
8 support accurate thresholds, threshold selection module 216 adopts pre-determined  
9 thresholds for use until the frames within the sliding window do not result in such  
10 a high standard deviation.

11 It is to be appreciated that use of an average and standard deviation of  
12 frame histograms is but one example of any of a number of alternate statistical  
13 tools which threshold selection module 216 could bring to bear to dynamically  
14 generate and verify thresholds for use in accordance with the teachings of the  
15 present invention. Any of a number of alternate statistical tools such as, for  
16 example, Analysis of Variance (ANOVA), test analysis, and the like may well be  
17 used to set and verify the veracity of a threshold without deviating from the spirit  
18 and scope of the present invention.

19 As introduced above, the flashlight detector 214 is selectively invoked to  
20 determine whether one or more frames, with a histogram difference between  
21 consecutive frames that exceeds some threshold (e.g.,  $T_b$ ) is the result of a  
22 flashlight or a shot transition. In general, flashlight detector 214 analyzes one or  
23 more attributes of a current frame against such attributes of frames in a sliding  
24 window to the right and to the left of the current frame to determine whether the  
25

light histogram variation is the result of a relatively isolated flashlight, or the result of a shot transition.

According to one example implementation, flashlight detector 214 generates a ratio of the light intensity of the current frame as compared to sliding windows of frames on either side of the current frame. More particularly, flashlight detector utilizes the average intensity difference (AID) measure to distinguish flashlight events from actual shot boundaries. Following are definitions of some metrics that are used in the shot detection method.

$$D_i = \sum_{j=1}^{Bins} |H_i(j) - H_{i-1}(j)| \quad (1)$$

$$AI_i = \sum_{j=1}^{Bins} j * H_i(j) / \sum_{j=1}^{Bins} H_i(j) \quad (2)$$

$$AI_{i-1} = \sum_{j=1}^{Bins} j * H_{i-1}(j) / \sum_{j=1}^{Bins} H_{i-1}(j) \quad (3)$$

$$AID_i = AI_i - AI_{i-1} \quad (4)$$

Where  $H_i(j)$  indicates the histogram value for gray level  $j$  in frame  $i$ .  $Bins$  is the bin number for the histogram, generally we choose 256 for gray level histogram. For a color image, each bin represents the intensity of a color. Typically, there are 256 bins for each color component (Red, Green, and Blue) in a color image. For a gray image, each bin represents an intensity of gray-level. Typically, there are 256 bins for a gray image (i.e., one bin for each of the 256 gray levels).  $D_i$  denotes the histogram difference between frame  $i$  and its preceding frame ( $i-1$ ).  $AI_i$  is the average intensity value of the frame  $i$ , and  $AID_i$  is the average intensity difference between frame  $i$  and frame ( $i-1$ ).

According to one implementation, introduced above, the metric  $D$  (equation 1) is compared to the high threshold,  $T_b$ , by shot boundary detector 212 to decide whether to invoke flashlight detector module 214, while metric  $AID$  (equation 4) is used by flashlight detector module 214 to distinguish a shot boundary from a flashlight event. To distinguish a shot cut from a flashlight event, flashlight detector module 214 compares the result of the average intensity analysis, above, to a cut model and a flashlight model.

Turning briefly to Fig. 3, a graphical illustration of an example ideal cut model and ideal flashlight event model is presented, according to one embodiment of the present invention. In accordance with the illustrated example illustration of Fig. 3, a shot cut model 300 is presented as distinguished from the flashlight event model 320, according to one embodiment of the present invention. When a true abrupt transition occurs (304, 310), i.e., a shot cut, the average intensity changes from one level (302, 308) to another level (306, 312) and this change will sustain itself for at least a relatively long period during the next whole shot. Alternatively, when a flashlight occurs (324, 328), the average intensity changes from one level (322, 326) to another level, and then fall back to the original level (322, 326), usually, within a couple of frames, which is significantly less than that of the true cut case.

Returning to Fig. 2, flashlight detector 214 defines two parameters (or, ‘heights’) used to classify a frame in which an abrupt change in intensity occurs:

- 22           **H1:** the average intensity difference between current  
23                 frame and previous frame
- 24           **H2:** the average intensity difference between frames in  
25                 left sliding window preceding the current frame and

frames in right sliding window after the current frame.

In the ideal flashlight model (320 of Fig. 3), the H2 parameter goes to zero (0) because the average intensity of frames preceding the current frame and the average intensity of frames after the current frames are at the same level (e.g., 322, 326) within the same shot. Alternatively, in the ideal cut model (300 of Fig. 3), the H2 parameter is substantially identical to the H1 parameter because the average intensity of frames preceding the current frame (302, 308) is not at the same level with that of frames (306, 312) after the current frames.

Accordingly, flashlight detector module 214 calculates a ratio of the H1 and H2 parameters, in accordance with equation (5) below:

$$Ratio = H1 / H2 \quad (5)$$

According to one implementation, the H2 parameter is calculated using the average intensities of 5-7 frames preceding current frame (i.e., left sliding window of frames), and the average intensities of 5-7 frames after the current frame (right sliding window). It will be appreciated that sliding windows of more or less frames could well be used, the size of 5-7 frames described herein is chosen because flashlight events do not, generally, last longer than 5-7 frames. According to one example implementation, flashlight detector module 214 reduces computational complexity by utilizing only a subset of the frames within each sliding window to generate the average intensity (AI) for the sliding window. According to one implementation, flashlight detector 214 utilizes the minimal two of all frames within the sliding window as the average intensity level for the

sliding window. By using only the minimal two of all the frames within the sliding window, the effect of the flashlight on the average is reduced, and one or two statistically unstable (in terms of light intensity) frames are omitted from the calculation.

As Ratio goes to a value of one (1), flashlight detector 214 concludes that the intensity change is due to a shot cut event and is, therefore indicative of a shot boundary. Deviations from a Ratio value of one are determined to be indicative of a flashlight event.

As used herein, storage/memory 206 and input/output interface(s) 208 are each intended to represent those elements as they are well known in the art. Storage/memory 206 is utilized by media analysis agent 104 to maintain, at least temporarily, media content 218 and detected shots 220. The I/O interface(s) 208 enable media analysis agent 104 to communicate with external elements and systems, facilitating a distributed architecture and remote operation.

Application(s) 210 are intended to include a wide variety of application(s) which may use, or be used by, media analysis engine 204. In this regard, application(s) 210 may well include a graphical user interface (GUI), a media player, a media generator, a media database controller, and the like.

Given the foregoing, it is to be appreciated that media analysis agent may well be implemented in a number of alternate embodiments. According to one implementation, media analysis agent 104 is implemented in software as a stand-alone application, as a subset of a higher-level multimedia application such as, for example, a media decoder application, a media rendering application, a browser application, a media player application, and the like. Alternatively, media analysis agent 104 may well be implemented in hardware, e.g., in an application specific

integrated circuit (ASIC), a controller, a programmable logic device (PLD), in a multimedia accelerator peripheral, and the like. Such alternate implementations are anticipated within the scope and spirit of the present invention.

### **Example Data Structure**

**Fig. 4** graphically illustrates an example data structure within which detected shots are stored, according to one embodiment of the present invention. As introduced above, once shot boundary detector 212 and/or flashlight detector module 214 have identified a shot boundary, the frames accumulated by shot boundary detector 212 since the last identified shot boundary and the currently identified shot boundary are segmented into a uniquely identifiable shot within memory 216. More particularly, the frames comprising such shots are initially stored in memory 220. In accordance with the illustrated example embodiment of Fig. 4, memory 206 is depicted comprising a subset (220) within which is stored a plurality of frames (402-406) associated with a detected shot. While one shot is currently depicted, it is to be appreciated that the size of detected shot memory 220 increases to service the number of shots detected by media analysis engine 204.

Once the media analysis agent 104 has completed segmenting received media content, controller 202 may well promote the detected shots from memory 220 to another storage location, e.g., a mass storage device, etc.

### **EXAMPLE OPERATION AND IMPLEMENTATION**

Having introduced the operating environment and functional elements of media analysis agent 104 with reference to Figs. 1-4, above, the operation of the system will now be developed more fully below with reference to Figs. 5-7, below. For

1 ease of illustration, and not limitation, the operation of media analysis agent 104  
2 will be developed below in the context of segmenting video content. However, as  
3 introduced above, the teachings of the present invention may well be adapted to  
4 segmenting other types of media content such as, for example, audio content.

5 **Fig. 5** is a flow chart of an example method of shot boundary detection,  
6 according to one embodiment of the present invention. In accordance with the  
7 illustrated example implementation of Fig. 5, the method begins with block 502  
8 wherein media analysis agent 104 receives an indication to segment media content  
9 into shots. In response, media analysis agent 104 selectively invokes an instance  
10 of shot boundary detector 212 of media analysis engine 204 to parse media content  
11 into uniquely identifiable shots. According to one implementation, introduced  
12 above, shot boundary detector 212 utilizes a multi-threshold boundary detection  
13 scheme to facilitate detection of both abrupt boundaries as well as gradually  
14 occurring boundaries.

15 In block 504, to facilitate the accurate detection of shot boundaries, shot  
16 boundary selection module 212 invokes an instance of adaptive threshold selection  
17 module 216 to calculate a high threshold ( $T_b$ ) and a low threshold ( $T_s$ ) for use in  
18 the multi-threshold detection scheme of the shot boundary detector 212. As  
19 introduced above, and developed more fully below, adaptive threshold selector  
20 module 216 calculates proposed threshold values ( $T_b$ ,  $T_s$ ) based on one or more  
21 attributes of frames in a sliding window of frames adjacent to a current frame. An  
22 example method for threshold selection is developed more fully below, with  
23 reference to Fig. 8.

24 In block 506, shot boundary detector 212 determines whether a change in  
25 light intensity between adjacent frames exceeds a threshold value. More

particularly, as introduced above, shot boundary detector 212 calculates a difference in intensity histograms between two adjacent frames ( $D_i$ ), in accordance with equation 1, above. This difference is then compared to the high threshold  $T_b$ . If the difference in light intensity histograms ( $D_i$ ) exceeds the high threshold, shot boundary detector 212 concludes that there has been an abrupt change in light intensity across adjacent frames and selectively invokes an instance of the flashlight detector module 214 to determine whether the abrupt change was caused by an abrupt shot boundary (e.g., caused by a shot cut), or whether the change was caused by a flashlight event, block 508. An example method for distinguishing a shot cut from a flashlight event is presented in more detail below, with reference to Fig. 6.

If, in block 506, the histogram difference does not exceed the high threshold, shot boundary detector 212 determines whether it reaches a lower threshold,  $T_s$ , block 510. As introduced above, the lower threshold ( $T_s$ ) is set as an indicator of a potential gradual transition between shots, e.g., a fade, wipe, etc. If the histogram difference does not exceed the low threshold,  $T_s$ , shot boundary detector 212 concludes that the frame does not represent a boundary condition, and the analysis continues with the next frame in the sequence, block 512.

If, in block 510, the histogram difference does exceed the lower threshold,  $T_s$ , shot boundary detector 212 adds the histogram difference to an Accumulated Difference (AD) for a number of frames ( $N$ ), block 514. The AD is then compared to the high threshold,  $T_b$ , block 516. If the accumulate difference exceeds the high threshold,  $T_b$ , shot boundary detector 212 concludes that a gradual shot boundary has been reached, block 518 and the frames comprising the shot are added to the detected shot memory 220.

If, the AD does not exceed the high threshold, Tb, in block 516, the process continues with block 512 with analysis of the next frame in the sequence.

It is to be appreciated that the multi-threshold shot boundary detection scheme introduced above facilitates detection of both hard, abrupt shot boundaries and gradual shot boundaries utilizing dynamically determined, and verified thresholds that are appropriately sized given the content of the media analyzed. Moreover, the multi-threshold shot boundary detection scheme distinguishes abrupt deviations in light intensity that are due to shot cut from those due to flashlight events within and across shot boundaries, thereby improving shot boundary detection performance by eliminating the false-positive identification often associated with flashlight events.

**Fig. 6** is a flow chart of an example method of distinguishing a cut model from a flashlight model, according to one aspect of the present invention. It is to be appreciated that although introduced in the context of the innovative multi-threshold boundary detection scheme introduced above, the method of Fig. 6 may well be used in conjunction with other shot boundary detection schemes.

In accordance with the illustrated example embodiment of Fig. 6, the method of block 508 (of Fig. 5) begins with blocks 602 and 604. In block 602, flashlight detector module 214 identifies the first and second minimal average intensity values (Min 11 and Min 12) in a left sliding window of frames. Similarly, in block 604, flashlight detector module 214 identifies the first and second minimal average intensity values (Min 21 and Min 22) in a right sliding window of frames. As introduced above, the left sliding window of frames may comprise the 5-7 frames prior to (left sliding window), and subsequent to (right sliding window) the current frame from which the minimal average intensity values are chosen.

In block 606, flashlight detector 214 calculates the average intensity (AI) for each of the sliding windows utilizing the minimal two average intensities of each window identified in blocks 602 and 604, respectively. The absolute value of the difference in average intensity of the left sliding window and the right sliding window is calculated and assigned to the parameter H1. In addition, flashlight detector module 214 calculates the average intensity change of the current frame, H2, as well as the Ration of H1 to H2 (per equation 5, above).

In block 608 flashlight detector module determines whether the calculated Ration (eqn. 5) is greater than a threshold indicator. According to one implementation, a ratio that varies from a value of one (1) is an indicator of a flashlight event. Thus, if the ratio does not exceed a threshold, flashlight detector 214 concludes that the ratio is indicative of a flashlight event, not a shot boundary, and processing continues with block 512 of Fig. 5.

If flashlight detector 214 determines that the ratio does exceed the set threshold, it concludes that a shot boundary has been detected, block 612. Accordingly, shot boundary detector 212 stores the frames comprising the identified shot in the detected shot memory 220. Upon detecting a shot boundary and storing the frames comprising the shot, shot boundary detector 212 sets the accumulated difference (AD) value to zero (0), and continues the analysis with the next frame in the sequence of frames, if any remain.

**Fig. 7** is a flow chart of an example method of dynamically generating an adaptive threshold(s) to identify shot boundaries in a video sequence, according to one aspect of the present invention. As above, it is to be appreciated that although depicted in accordance with the multi-threshold shot boundary detection scheme,

1 the method of Fig. 7 may well be applied to other means of detecting shot  
2 boundaries and is not limited to the example implementation.

3 In accordance with the illustrated example implementation of Fig. 7, the  
4 method of block 504 (Fig. 5) begins with block 702, where adaptive threshold  
5 selection module 216 calculates an average and standard deviation values of the  
6 histogram difference(s) in a sliding window. As introduced above, the sliding  
7 window may well be 20-25 frames, although windows of greater or less frames  
8 may well be used. From the calculated average intensity difference (AID) of  
9 histograms in the sliding window, adaptive threshold selection module 216  
10 generates a proposed set of thresholds. According to one implementation, the  
11 proposed low threshold is 2-3 times the average, while the proposed high threshold  
12 is 4-5 times the average.

13 In block 704, adaptive threshold selection module 216 determines whether  
14 the standard deviation of the average intensity values of the sliding window reach  
15 a certain threshold. As introduced above, the higher the standard deviation, the  
16 more unreliable the thresholds will be. Accordingly, the standard deviation  
17 threshold is set to ensure the calculated low- and high threshold values used in the  
18 boundary detection provide accurate results. If the standard deviation meets or  
19 exceeds the standard deviation threshold, the proposed low and high threshold  
20 values ( $T_s$ ,  $T_b$ ) are abandoned in favor of a previous threshold, block 706.

21 If, however, the standard deviation does not reach or exceed the standard  
22 deviation threshold, the proposed threshold values of  $T_s$  and  $T_b$  are introduced for  
23 use in the multi-threshold boundary selection scheme of Fig. 5, block 708.

1           **ALTERNATE EMBODIMENT(S)**

2           **Fig. 8** is a block diagram of a storage medium having stored thereon a  
3           plurality of instructions including instructions to implement the teachings of the  
4           present invention, according to yet another embodiment of the present invention.  
5           In general, Fig. 8 illustrates a storage medium/device 800 having stored thereon a  
6           plurality of executable instructions including at least a subset of which that, when  
7           executed, implement the media analysis agent 104 of the present invention.

8           As used herein, storage medium 800 is intended to represent any of a  
9           number of storage devices and/or storage media known to those skilled in the art  
10          such as, for example, volatile memory devices, non-volatile memory devices,  
11          magnetic storage media, optical storage media, and the like. Similarly, the  
12          executable instructions are intended to reflect any of a number of software  
13          languages known in the art such as, for example, C++, Visual Basic, Hypertext  
14          Markup Language (HTML), Java, eXtensible Markup Language (XML), and the  
15          like. Moreover, it is to be appreciated that the storage medium/device 800 need  
16          not be co-located with any host system. That is, storage medium/device 800 may  
17          well reside within a remote server communicatively coupled to and accessible by  
18          an executing system. Accordingly, the software implementation of Fig. 8 is to be  
19          regarded as illustrative, as alternate storage media and software embodiments are  
20          anticipated within the spirit and scope of the present invention.

21          Although the invention has been described in language specific to structural  
22          features and/or methodological steps, it is to be understood that the invention  
23          defined in the appended claims is not necessarily limited to the specific features or  
24          steps described. For example, the inventive concepts presented herein may well be  
25          used to identify distinct audio content (e.g., songs) on a storage medium populated

1 with a plurality of such audio content (e.g., a music CD). In accordance with this  
2 alternate implementation, an application 210 of media analysis agent 104 generates  
3 a light histogram representation of the audio content. Any of a number of  
4 techniques may well be used to perform this audio to visual transform such as, for  
5 example, spectral analysis and the like. Once transformed, media analysis agent  
6 104 proceeds, as described above, to parse the audio content into uniquely  
7 identifiable segments. Thus, it is to be appreciated that the specific features and  
8 steps are disclosed as but an example implementation of the broader inventive  
9 concepts introduced herein.